Titre: Title:	Bisimulation-based non-deterministic admissible interference and its application to the analysis of cryptographic protocols
Auteurs: Authors:	Stéphane Lafrance, & John Mullins
Date:	2002
Type:	Communication de conférence / Conference or Workshop Item
Référence: Citation:	Lafrance, S., & Mullins, J. (2002, January). Bisimulation-based non-deterministic admissible interference and its application to the analysis of cryptographic protocols [Paper]. Computing: the Australasian Theory Symposium (CATS 2002), Monash University, Melbourne, Australia (24 pages). Published in Electronic Notes in Theoretical Computer Science, 61. https://doi.org/10.1016/s1571-0661(04)00311-1

Nom de la conférence: Conference Name:	Computing: the Australasian Theory Symposium (CATS 2002)
Date et lieu: Date and Location:	2002-01-28 - 2002-02-01, Monash University, Melbourne, Australia
Maison d'édition: Publisher:	Elsevier
URL officiel: Official URL:	https://doi.org/10.1016/s1571-0661(04)00311-1
Mention légale: Legal notice:	

URL: http://www.elsevier.nl/locate/entcs/volume61.html 24 pages

Bisimulation-based Non-deterministic Admissible Interference and its Application to the Analysis of Cryptographic Protocols

Stéphane Lafrance [1,2] John Mullins [1,3],

Dept. of Computer and Software Engineering
École Polytechnique de Montréal
Montréal, Canada

Abstract

In this paper, we first define *bisimulation-based non-deterministic admissible interference* (BNAI), derive its process-theoretic characterization and present a compositional verification method with respect to the main operators over communicating processes, generalizing in this way the similar trace-based results obtained in [19] into the finer notion of *observation-based bisimulation* [6]. Like its trace-based version, BNAI admits information flow between secrecy levels only through a downgrader (e.g. a cryptosystem), but is phrased into a generalization of *observational equivalence* [18]. We then describe an admissible interference-based method for the analysis of cryptographic protocols, extending, in a non-trivial way, the non interference-based approach presented in [11]. Confidentiality and authentication for cryptoprotocols are defined in terms of BNAI and their respective bisimulation-based proof methods are derived. Finally, as a significant illustration of the method, we consider simple case studies: the paradigmatic examples of the Wide Mouthed Frog protocol [1] and the Woo and Lam one-way authentication protocol [25]. The original idea of this methodology is to prove that the intruder may interfere with the protocol *only* through selected channels considered as admissible when leading to harmless interference.

1 Introduction

One of the basic concerns in systems analysis is to ensure that programs do not leak sensitive data to a third party, either maliciously or inadvertently.

[1] The first author has been supported by an NSERC scholarship for graduate studies and the second, by the NSERC grant no. 138321-01 from the Canadian Goverment.
[2] Email: stephane.lafrance@polymtl.ca
[3] Email: John.Mullins@polymtl.ca

This key aspect of security concerns is often referred to as secrecy. *Information flow analysis* addresses this concern by clarifying conditions when a flow of information in a program is safe (i.e. high-level information never flows into low-level channels). These conditions, called *non interference* properties [10], capture any causal dependency between high-level actions and low-level behavior.

However, many practical secrecy problems go beyond the scope of non interference. Cryptosystems, for example, permit encrypted private or classified information to flow safely onto unprotected (i.e. low-level) channels despite the obvious causal dependency between the secret data m and encryption key K, on the one hand, and, the declassified data $\{m\}_K$ (m encrypted by K), on the other. Indeed, any variation of m or K is reflected in $\{m\}_K$. In this case, the main concern is to ensure that programs leak sensitive information *only* through the cryptosystem or, more generally, through the downgrading system. *Admissible interference* [19] is such a property. In this paper, we define bisimulation-based semantics for non-deterministic admissible interference. It appears that *observation-dependent bisimulation* based on an observation criterion $\mathcal{O}$ or $\mathcal{O}$-bisimulation (called $\mathcal{O}$-congruence in [6]) provides a suitable theoretical framework for expressing *bisimulation-based non-deterministic admissible interference* (BNAI). As we shall see, BNAI has an elegant process-theoretic characterization (traditionally called the unwinding theorem in the theory of information flow) and attractive compositionality properties.

Non interference-based methods have been designed to analyze cryptographic protocols [12,9]. The basic idea of the method is to prove that no intruder can interfere with the protocol. In this paper, we refine this method by considering as admissible the interference caused by encryption. This admissible interference can be expressed by simply identifying downgrading actions corresponding to encryption actions occurring in the protocol. This paper will highlight two kinds of advantages of the admissible interference-based method over a non interference-based one. In some cases, the method permits analysis of the protocol's information flow without the necessity of extending the syntax of the process algebra with encryption and decryption operators. In other cases, it allows harmless interference, i.e. interference that does not correspond to a successful attack, to be discarded at the specification level, rather than screening it manually from the by-products of the verification process.

The paper is organized as follows. A variant of the value-passing CCS, extended with Boudol's observation criteria and its observation-dependent bisimulation-based semantics, is introduced in section 2. Non-deterministic admissible interference based on observation-dependent bisimulation is presented in section 3 with its algebraic process characterization and its compositionality properties with respect to the main process operators. In section 4, we present different ways to use BNAI in the analysis of cryptoprotocols. More particularly, we focus on confidentiality and authentication properties. These properties are defined in terms of BNAI and their respective bisimulation-

based proof methods are derived. The method is further investigated through the Wide Mounted Frog protocol in section 5 and the Woo and Lam one-way authentication protocol in section 6. We conclude in section 7 with an overview of related and future works.

2 Preliminaries

2.1 Value-passing CCS

We need to start the discussion by identifying a computational syntax to structure the investigation around. Our work is based on *value-passing CCS* [18] modified in various ways as we move along.

We consider the following message algebra, whose *terms*, ranged over by a, are defined by:

$$a := \quad v \text{ (value)} \mid x \text{ (variable)} \mid (a, a) \text{ (pair)} \mid \{a\}_a \text{ (encryption)}.$$

We denote $fv(a)$ the set of (free) variables appearing in a and we say that a is a *closed term* when $fv(a) = \emptyset$. Throughout this paper, any closed encryption term $\{a_1\}_{a_2}$ is viewed as the atomic value resulting from the encryption of the closed term a_1 using the closed term a_2 as key. For any (atomic) value v and $x \in fv(a)$, we write $a[v/x]$ to denote the setting of every occurrence of x in a to value v and $a[v_1/x_1][v_2/x_2]$ is noted as $a[v_1/x_1, v_2/x_2]$, and so on. Further, we assume a set of at most denumerable *channels*, ranged over by c. Every channel is typed, i.e. has a unique structure of terms (messages) that can be sent and received over it. We write $\mathrm{dom}(c)$ to denote the domain of terms that can be carried along c.

Actions of our extended value-passing CCS, ranged over by μ, are obtained from combinations of one channel and one term, as follows:

- $\overline{c(a)}$ or $\overline{c}(a)$ (*output action*),
- $c(a)$ (*input action*),
- τ (*internal action*).

for any $a \in \mathrm{dom}(c)$. Thus, the set of $Act = Vis \cup \{\tau\}$ contains a set of *visible actions* $Vis = In \cup Out$, where In is a set of *input actions*, $Out = \overline{In}$ is a set of *output actions* and the function $\overline{[.]} : Vis \to Vis$ is such that $\overline{\overline{\mu}} = \mu$. We define the set of free variables of an action μ, denoted by $fv(\mu)$, as the set $fv(a)$ if $\mu = \overline{c}(a)$ or $\mu = c(a)$, and $fv(\mu) = \emptyset$ if $\mu = \tau$. We say that an action μ is *closed* if $fv(\mu) = \emptyset$, otherwise we say that it is *open*, and we use α to range over the set of closed actions.

Agents (ranged over by P and Q) are constructed as follows:

- $\mathbf{0}$ (*empty agent*);
- $\mu.P$ (*prefix*);
- $P[v/x]$ (*assignment*);
- $P + Q$ (*sum*);

- $P|Q$ (*parallel composition*);
- $P\backslash L$ (*restriction*);
- $[x_1 = x_2]\, P$ (*match*);
- $P/\mathcal{O}$ ($\mathcal{O}$-*observation*);

where v is a value, x, x_1, x_2 are variables, L is any set of actions and $\mathcal{O}$ is a partial mapping from Act^* to Act called *observation criterion* and whose intended meaning will be clarified in the next section. With this syntax, recursion is dealt with by using agent names, e.g. by defining $P = \mu_1.P'$ and $P' = \mu_2.P$ for the $\mu_1.\mu_2$ loop. We define $fv(P)$, the set of *free variables* occurring in P, as the set of variables x appearing in P and not in the scope of an input prefix μ such that $x \in fv(\mu)$. When $x \in fv(P)$, we often write $P(x)$ (with $P(x_1)(x_2) = P(x_1, x_2)$, etc.) and $P(v)$ instead of $P[v/x]$ (where every free occurrence of x in P is set to v). Otherwise, the variable x is said to be *bound*. A *closed agent*, or simply a *process*, is an agent P such that $fv(P) = \emptyset$. For the sake of simplicity, we often omit writing $\mathbf{0}$ by using the notation "α" instead of "$\alpha.\mathbf{0}$".

We shall now define a downgrading process as an extension of a process to model systems and computations of computing entities interacting at different trust levels in an environment controlled by a downgrading system. A *downgrading process* is then a process whose set of visible actions Vis is a partition of three sets Lo, Hi and Dwn such that $\overline{Lo} = Lo$, $\overline{Hi} = Hi$ and $\overline{Dwn} = Dwn$.

2.2 Observation Criterion

In [6], Boudol has defined the notion of *observation criterion* to express an observation of actions with the aim of considering the equivalence between processes. Such a criterion on a set A of actions defines a set B of *observables* or *experiments*. In this paper, only observation criteria of Act^* are considered.

Definition 2.1 An *observation criterion* of Act^* is a partial mapping $\mathcal{O}$ from Act^* to Act.

The intended meaning is that all sequences of actions in $\mathcal{O}^{-1}(\alpha)$ are held to carry out the same observation α. Thus, it is natural not to require the mapping to be total: some sequences may be invisible or meaningless from a given point of view. We are particularly interested in the observation criterion $\mathcal{O}_{Hi}$ defined by

$$\mathcal{O}_{Hi}^{-1}(\alpha) = \begin{cases} \tau^*\ \alpha\ \tau^* & \text{if } \alpha \in Vis \\ \tau^* & \text{if } \alpha = \tau \end{cases}$$

and the observation criterion $\mathcal{O}_{Lo}$ defined by

$$\mathcal{O}_{Lo}^{-1}(\alpha) = \begin{cases} (\{\tau\} \cup Hi)^*\ \alpha\ (\{\tau\} \cup Hi)^* & \text{if } \alpha \in Lo \cup Dwn \\ (\{\tau\} \cup Hi)^* & \text{if } \alpha = \tau. \end{cases}$$

Thus, two sequences are equivalent through the weak criterion $\mathcal{O}_{Hi}$ if their visible content is the same and two sequences are equivalent through the criterion $\mathcal{O}_{Lo}$ if their visible low-level content is the same.

2.3 Semantics

The operational semantics of a process obtained from this language can also be viewed as an extension of the usual notion of a non-deterministic finite-state automaton where we allow an infinite set of states and where we generally do not consider final states. Let c be a channel, let $a \in \mathrm{dom}(c)$ be such that $fv(a) = \{x_1, \ldots, x_n\}$ and let $v, v_1, \ldots, v_n$ be values. Let also α be a closed action, γ a sequence of closed actions, $L \subseteq Vis$ and P, P', Q and Q' agents. The semantics of processes is defined as follows:

Prefix
$$\frac{-}{\alpha.P \xrightarrow{\alpha} P}$$

Input
$$\frac{-}{c(a).P \xrightarrow{c(a[v_1/x_1,\ldots,v_n/x_n])} P[v_1/x_1,\ldots,v_n/x_n]}$$

Sum
$$\frac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'} \quad \text{and} \quad \frac{Q \xrightarrow{\alpha} Q'}{P+Q \xrightarrow{\alpha} Q'}$$

Parallel
$$\frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q} \quad \text{and} \quad \frac{Q \xrightarrow{\alpha} Q'}{P|Q \xrightarrow{\alpha} P|Q'}$$

Synchronization
$$\frac{P \xrightarrow{\overline{c}(a)} P' \quad \text{and} \quad Q \xrightarrow{c(a)} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$$

Restriction
$$\frac{P \xrightarrow{\alpha} P' \quad \text{and} \quad \alpha \notin L \cup \overline{L}}{P\backslash L \xrightarrow{\alpha} P'\backslash L}$$

Match
$$\frac{P \xrightarrow{\alpha} P'}{[v=v]\ P \xrightarrow{\alpha} P'}$$

$\mathcal{O}$ − Observation
$$\frac{P \xrightarrow{\gamma} P' \quad \text{and} \quad \mathcal{O}(\gamma)=\alpha}{P/\mathcal{O} \xrightarrow{\alpha} P'/\mathcal{O}}$$

where notation $P \xrightarrow{\gamma} P'$ stands for a *computation* of the sequence of closed actions $\gamma = \alpha_0\alpha_1 \ldots \alpha_n \in Act^*$ in the process P i.e. the finite string of transitions satisfying $P \xrightarrow{\alpha_0} P_1 \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_n} P'$. Given a process P and an observation criterion $\mathcal{O}$, we say that $P/\mathcal{O}$ is the $\mathcal{O}$-*observation* of P. The notion of the $\mathcal{O}$-observation of a process is aimed at defining the process on an observable

resulting from its observation through the observation criterion $\mathcal{O}$.

Example 2.2 Consider the observation criterion $\mathcal{O}_{Hi}$ and $\mathcal{O}_{Lo}$ previously defined. Put $Hi = \{\alpha\}$ and $Lo = \{\beta_1, \beta_2, \beta_3\}$. Let P be a process having the semantics illustrated in Fig. 1. Then the semantics of processes $P/\mathcal{O}_{Hi}$ and $P/\mathcal{O}_{Lo}$ are given in Fig. 2. Note that, in both systems, we omitted looping τ transitions at every state.

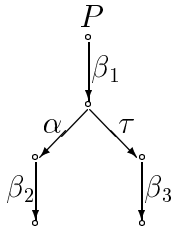

Fig. 1. Semantics of process P.

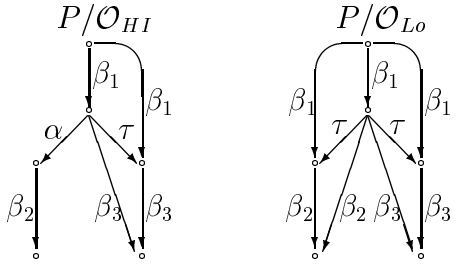

Fig. 2. Semantics of processes $P/\mathcal{O}_{Hi}$ and $P/\mathcal{O}_{Lo}$.

We say that agent P' is *reachable* from P, also called a *derivative*, if there is a computation $P \xrightarrow{\gamma} P'$ for some $\gamma \in Act^*$. We shall frequently make use the set $\mathcal{R}(P) = \{P' \mid \exists_{\gamma \in Act^*} P \xrightarrow{\gamma} P'\}$ as the set of reachable agents from P.

2.4 Observation-dependent Bisimulation

The concept of $\mathcal{O}$-bisimulation [4] captures the notion of behavioral indistinguishability through $\mathcal{O}$.

Definition 2.3 (i) Given processes P and Q and observation criterion $\mathcal{O}$, an $\mathcal{O}$-simulation of P by Q is a relation $R \subseteq \mathcal{R}(P) \times \mathcal{R}(Q)$ such that
- $(P, Q) \in R$,
- If $(P_1, Q_1) \in R$ and $P_1 \xrightarrow{\alpha} P_2$, then there exists $Q_2 \in \mathcal{R}(Q)$ such that $(P_2, Q_2) \in R$ and $Q_1 \xrightarrow{\gamma} Q_2$ with $\mathcal{O}(\gamma) = \mathcal{O}(\alpha)$.

In such a case, we denote $P \sqsubseteq_{\mathcal{O}} Q$.

(ii) An $\mathcal{O}$-simulation R of P by Q is an $\mathcal{O}$-*bisimulation* if R^{-1} is an $\mathcal{O}$-simulation of Q by P. We say that P and Q are $\mathcal{O}$-*bisimilar* (denoted $P \approx_{\mathcal{O}} Q$) in the case where they are related by an $\mathcal{O}$-*bisimulation*.

[4] Called $\mathcal{O}$-congruence in [6]

The best known examples of this are the criteria defining strong and weak bisimulations of CCS. Both are special cases of criteria obtained from projections. Two sequences are equivalent through the weak criterion $\mathcal{O}_{Hi}$ if their visible content is the same. When $\mathcal{O}$ is the identity on Act, one gets the strong criterion through which each sequence of actions is observable and distinguishable from any other sequence. In this way, weak bisimulation could be seen as $\mathcal{O}_{Hi}$-bisimulation, and (strong) bisimulation could be seen as $\mathcal{O}_{Act}$-bisimulation where $\mathcal{O}_{Act}$ is the identity observation criterion, i.e. $\mathcal{O}_{Act}(\alpha) = \alpha$ for every $\alpha \in Act$. We denote (strong) bisimulation between two processes P and Q simply with $P \approx Q$ and (strong) simulation of P by Q with $P \sqsubseteq Q$. More generally, the concept of $\mathcal{O}$-bisimulation is related to bisimulation in the following way.

Proposition 2.4 *Given processes P and Q and observation criterion $\mathcal{O}$, we have*

$$P \approx_{\mathcal{O}} Q \quad \textit{if and only if} \quad P/\mathcal{O} \approx Q/\mathcal{O}.$$

It is important to note that Prop. 2.4 still holds when $\mathcal{O}$-bisimulation and bisimulation are both replaced with $\mathcal{O}$-simulation and simulation.

3 Bisimulation-based Non-deterministic Admissible Interference

A drastic solution to avoid interference of high-level users on low-level users, which is causing a very typical problem in computer security, is to forbid these possible interferences. Several definitions of non interference have been proposed in the literature (see [17] for a survey). In [10], a trace-based generalization of non interference, called *strong non-deterministic non interference* (SNNI), has been proposed. It is satisfied when that the low-level visible content of any system behavior, namely a visible trace, is still a system behavior. Non-deterministic admissible interference (NAI) has been introduced in [19]. It is a trace-based property requiring SNNI everywhere but through dedicated downgrading channels. The main result of this paper is the introduction of bisimulation-based non-deterministic admissible interference (BNAI) that exploits the concept of observation-dependent bisimulation presented in section 2.4. This section also gives an algebraic characterization of BNAI through an Unwinding Theorem (Theorem 3.3) and results on compositionality w.r.t. the main constructors of CCS (Theorem 3.4).

3.1 Semantics

In order to gain a better understanding of BNAI, we introduce a bisimulation-based non interference property that refines SNNI and has suitable compositional properties. The following formulation of non interference requires that a process $\mathcal{O}_{Hi}$-simulates its $\mathcal{O}_{Lo}$-observation. Thus, roughly speaking,

bisimulation-based strong non-deterministic non interference (BSNNI) states that any low-level observable behavior has to be also a high-level process behavior, in order to disallow any correlation between a high-level behavior and a low-level observation.

Definition 3.1 Process P satisfies BSNNI if

$$P/\mathcal{O}_{Lo} \sqsubseteq_{\mathcal{O}_{Hi}} P.$$

It is not difficult to prove, using Prop. 2.4, that this property coincides with bisimulation-based strong non-deterministic non interference as proposed in [10].

Intransitive non interference refers to the information flow properties that require that systems admit information flow from the high level to the low level only through specific downgrading channels. To capture this property, it was proposed in [13] that any agent P' derived from P and executing no downgrading action be required to satisfy non interference. More precisely, for P to satisfy intransitive non interference $P' \setminus Dwn$ must satisfy non interference for every $P' \in \mathcal{R}(P)$. Rephrasing it in the context of BSNNI as the non interference property yields the definition of BNAI.

Definition 3.2 Process P satisfies *bisimulation-based non-deterministic admissible interference* (BNAI) if

$$\forall_{P' \in \mathcal{R}(P)} \quad (P' \setminus Dwn)/\mathcal{O}_{Lo} \ \sqsubseteq_{\mathcal{O}_{Hi}} \ (P' \setminus Dwn).$$

The next theorem presents an algebraic characterization of BNAI based on $\mathcal{O}_{Lo}$-bisimulation.

Theorem 3.3 (Unwinding Theorem for BNAI) *The process P satisfies BNAI if and only if*

$$\forall_{P' \in \mathcal{R}(P)} \quad P' \setminus Dwn \ \approx_{\mathcal{O}_{Lo}} \ P' \setminus (Dwn \ \cup \ Hi).$$

The proof of Theorem 3.3 is presented in Appendix A.1.

3.2 A Compositional Proof Method

The next theorem establishes the compositionality of BNAI over closed agents with respect to the restriction operator and a weak form of compositionality of BNAI with respect to the concurrent operator.

Theorem 3.4 (Compositionality Theorem for BNAI) *Let $L \subseteq Act$.*

(i) *If process P satisfies BNAI, then $P \setminus L$ satisfies BNAI.*

(ii) *If processes P and Q may not synchronize on downgrading actions and both satisfy BNAI, then $P|Q$ satisfies BNAI.*

The proof of Theorem 3.4 is given in Appendix A.2. This result extends a similar result for NAI obtained in [19]. A direct proof that a process satisfies BNAI is, according to Theorem 3.3, to exhibit for each derivative P',

a $\mathcal{O}_{Lo}$-bisimulation starting from P'. When the transition system obtained from the semantics is finite, this can be done automatically. Many tools, including the *Edinburgh Concurrency Workbench* (CWB) [7], exploits efficient algorithms for checking bisimilarity between finite processes, as developed in [15]. We are currently designing and implementing at the *École Polytechnique de Montréal*, in collaboration with the *Université d'Orléans*, a tool to check whether a finite state downgrading process satisfies BNAI or not (available at `www.crac.polymtl.ca`). We plan to extend this tool to cope with infinite-state processes such as those defined by totally normed Basic Process Algebra (BPA) [14] or Pushdown Processes [23].

4 Using BNAI to analyze cryptographic protocols

In this section, we give non trivial illustrations of how BNAI can be used to detect flaws in security protocols. The main contribution of this section is a general information flow method using BNAI that refines Focardi and Gorrieri's methods for analyzing cryptoprotocols [9,11] where the authors have either to extend the syntax and semantics of CCS before proceeding with analysis or to filter out manually meaningless interference (from authentication point of view) resulting from the analysis. Improvements given by BNAI depend on the type of security property under study:

- in the case of confidentiality properties (see section 5), downgrading actions may be interpreted to counter the unavoidable but harmless interference caused by encryption, and thus the process algebra does not have to be extended with encryption and decryption primitives;
- in most of the other cases, particularly for the authentication properties (see section 6), downgrading actions may be used to detect actions causing interference, but not corresponding to successful attacks, before analysis, rather than after analysis.

As we shall see, BNAI provides a natural interpretation of the following confidentiality property for security protocols:

> *No enemy process interacting with the protocol can discriminate, in an inadmissible way, the protocol's behavior and the behavior of the protocol exchanging no confidential information.*

A second property for security protocols on authenticity can be interpreted in terms of BNAI as follows:

> *No enemy process can interfere in an inadmissible way with the protocol.*

We now undertake the task of formalizing those properties in the context of our process algebra. Given such a formalization, we are also interested to derive corresponding Unwinding Theorems to verify such properties.

In the sequel, we use the variable X to range over process names and variables (including tuples) $w, x, y, z, \ldots$ to range over value terms. A crypto-

protocol P involving principals $A_1, A_2, \ldots A_n$ (specified as processes in value-passing CCS) is viewed as the following:

$$P(x_1, x_2, \ldots x_n) = (\ A_1(x_1) \mid A_2(x_2) \mid \ldots \mid A_n(x_n)\ ) \setminus C \tag{1}$$

where C corresponds to the set of public actions used in P. The restriction over the set C can be viewed as a forced synchronization of actions made on public channels. Every confidential data exchanged on a public channel must be properly encrypted since we assume that such channels are insecure. We shall also make the hypothesis that any other action, i.e. not belonging to C, is executed on a secure channel.

An attack on P executed by an enemy process E is specified as the process:

$$P_E(x_1, x_2, \ldots x_n, x_E) = (\ A_1(x_1) \mid A_2(x_2) \mid \ldots \mid A_n(x_n) \mid E(x_E)\ ) \setminus C. \tag{2}$$

Eq. 2 clearly expresses the fact that attackers may intercept any message (closed term) sent out on a public channel.

In such a specification, each principal X has its own set of private actions, noted $Hi(X)$, and we use notation $Hi(X_1, X_2)$ to denote the set $Hi(X_1) \cup Hi(X_2)$, and so on. Hence, we have $Hi = \bigcup_X Hi(X)$ and $Lo = C$. It is important to note that the content of the disjoint sets Hi, Lo and Dwn is, as we shall see, case dependent. In general, we shall use the notation c_X to denote a private channel belonging to a principal X and simply c for a public channel. For any principal X, we consider the following natural observation criterion $\mathcal{O}_X$ describing the actions observable by X which is defined as follows:

$$\mathcal{O}_X^{-1}(\alpha) = \begin{cases} (Hi^c(X) \cup \{\tau\})^* \ \alpha \ (Hi^c(X) \cup \{\tau\})^* & \text{if } \alpha \in Hi(X) \cup Dwn \cup Lo \\ (Hi^c(X) \cup \{\tau\})^* & \text{if } \alpha = \tau \end{cases}$$

where $Hi^c(X) = Hi \setminus Hi(X)$. For any two principals X_1 and X_2, we may also consider the joint observation criterion $\mathcal{O}_{X_1,X_2}$ defined as one might expect.

Depending on the type of security property we wish to enforce, each set $Hi(X)$ may contain *encryption actions* and *decryption actions*. Encryption is viewed as the sequence of actions $\overline{e_X}(x, y).cipher_X(\{y\}_x)$ where output action $\overline{e_X}(x, y)$ signifies the encryption of term y using key x (e.g. by sending y and x to X's local encrypter), and input action $cipher_X(z)$ then creates a (bound) variable z corresponding to the resulting value (often referred to as the term $\{y\}_x$). Similarly, decryption is viewed as the sequence of actions $\overline{d_X}(x, \{y\}_x)).read_X(y)$ where output action $\overline{d_X}(x, z)$ signifies the decryption of the term z using the key x (e.g. by sending z and x to X's local decrypter), and input action $read_X(y)$ waits for the resulting term y.

4.1 Preservation of Confidentiality

The major concern of cryptoprotocols is keeping the confidentiality of classified information that needs to be sent over private channels. Attacks on such protocols take different forms, from direct attempts to steal an entire confi-

dential message to much more subtle attempts to detect exchanges of private data. In this paper, a confidentiality property is introduced which is very sensitive to any kind of inadmissible information flow leading to unwanted secrecy leaks.

This highlights the fact that the principal X can only see actions coming from either a public channel, i.e. from Lo, or its own set $Hi(X)$ of private actions. In the case of confidentiality properties, we are particularly interested in the observation criterion $\mathcal{O}_X$ when X is an enemy process interacting with the protocol. For the following confidentiality property, the set Dwn of downgrading actions corresponds to actions causing admissible declassification of information such as proposed by admissible interference. This type of action is mainly used to indicate the execution of an encryption action as $cipher_X(\{y\}_x)$.

Given any value m, let $Act(m)$ be the set of actions containing m non-encrypted in its term (e.g. $read_X(m)$ or $\overline{e_X}(k,m)$, but not $cipher_X(\{m\}_k)$), and let $\mathcal{O}_m$ be the observation criterion defined by

$$\mathcal{O}_m^{-1}(\alpha) = \begin{cases} (Act^c(m)\cup\{\tau\})^* \ \alpha \ (Act^c(m)\cup\{\tau\})^* & \text{if } \alpha \in Act(m) \cup Hi(E) \cup Dwn \cup Lo \\ (Act^c(m)\cup\{\tau\})^* & \text{if } \alpha = \tau \end{cases}$$

where $Act^c(m) = Hi \setminus (Act(m) \cup Hi(E))$.

Definition 4.1 (Preservation of Confidentiality) The protocol $P(m)$ *preserves the confidentiality of message* m if, for every enemy process E,

$$\forall_{P'_E \in \mathcal{R}(P_E)} \quad (P'_E(m) \setminus Dwn)/\mathcal{O}_E \ \sqsubseteq_{\mathcal{O}_m} \ P'_E(m) \setminus Dwn.$$

This property may be viewed as

$$\forall_{E\text{: enemy process}} \quad P_E(m) \text{ satisfies } BNAI.$$

However, we must note that preservation of confidentiality offers an altered interpretation of BNAI once an enemy process E is fixed, since not every action from $Hi \setminus Hi(E)$ is considered a high-level action, only those containing confidential information. This property of preservation of confidentiality is illustrated is section 5 using the *Wide Mouthed Frog protocol* [1].

Remark 4.2 We note the trivial fact that if P_E has a derivative P'_E that may perform an action from $Hi(E) \cap Act(m)$, which clearly corresponds to a successful attack since process E can see m, then protocol P does not preserve the confidentiality of message m since such a transition belonging to $(P'_E(m) \setminus Dwn)/\mathcal{O}_E$ may not be $\mathcal{O}_m$-simulated by process $P'_E(m) \setminus (Dwn \cup Act(m))$.

We can establish the following unwinding theorem for our confidentiality property inspired by the unwinding theorem for BNAI (Theorem 3.3).

Theorem 4.3 *Protocol* $P(m)$ *preserves the confidentiality of message* m *if and only if for every enemy process* E,

$$\forall_{P'_E \in \mathcal{R}(P_E)} \quad P'_E(m) \setminus Dwn \ \approx_{\mathcal{O}_E} \ P'_E(m) \setminus (Dwn \ \cup \ Act(m)).$$

The Proof of Theorem 4.3 is given in Appendix A.3.

Example 4.4 Consider the following simple protocol where two principals A and B sharing secret key k_{AB} want to exchange a secret binary message m:

$$\begin{aligned} A(k_{AB}, m) &= \overline{e_A}(k_{AB}, m).cipher_A(\{m\}_{k_{AB}}).\overline{c_1}(\{m\}_{k_{AB}}) \\ B(k_{AB}) &= c_1(x).\overline{d_B}(k_{AB}, x).read_B(y). \\ &\qquad (\ [\Pi_1(y) = 0]\ \overline{c_2}(0)\ +\ [\Pi_1(y) = 1]\ \overline{c_2}(1)\ ) \end{aligned}$$

where c_1 and c_2 are public channels, $cipher_A \in Dwn$ is a downgrading channel allowing the declassification of $\{m\}_{k_{AB}}$, and Π_1 is the last-bit projection (e.g. $\Pi_1(10010) = 0$).

This particular example has an obvious inadmissible confidentiality break since it leaks pieces of information about m's content (in this case its parity) without revealing m entirely. Such an attack on confidentiality may be pursued by the following enemy process:

$$E = c_2(z).(\ [z = 0]\ \overline{even_E}\ +\ [z = 1]\ \overline{odd_E}\ )$$

which can evaluate the parity of the exchanged secret message m.

Using theorem 4.3, we see that this particular protocol fails to preserve the confidentiality of m. Let $A'(k_{AB}, m) = \overline{c_1}(\{m\}_{k_{AB}}) \in \mathcal{R}(A(k_{AB}, m))$ and $P'_E = (\ A'(k_{AB}, m) \mid B(k_{AB}) \mid E\ ) \setminus \{c_1, c_2\} \in \mathcal{R}(P_E)$. Then we have

$$P'_E(m) \setminus Dwn\ \not\approx_{\mathcal{O}_E}\ P'_E(m) \setminus (Dwn\ \cup\ Act(m))$$

since

$$(\ P'_E(m) \setminus Dwn\ )/\mathcal{O}_E\ \approx\ \tau.\tau.\tau.(\ [\ \Pi_1(m)\ =\ 0\ ]\ \tau.even_E\ +\ [\ \Pi_1(y)\ =\ 1\ ]\ \tau.odd_E\ ),$$

while $(\ P'_E(m) \setminus (Dwn\ \cup\ Act(m))\ )/\mathcal{O}_E\ \approx\ \tau.\tau$.

4.2 Preservation of Authenticity

An authentication protocol is a security protocol where a principal A wants to authenticate a second principal B and/or authenticate himself for B. Successful attacks on such protocols generally take the form of an enemy process convincing B that he is A. In many cases, A initiated the protocol with that enemy process which uses information obtained from A to execute his masquerade toward B, but an enemy process may also use information intercepted from public channels, as in section 5.

In order to work with authentication protocols, we adapt our notation established in Eq. 1 and Eq. 2. Thus, an authentication protocol where agent A initiates the authentication procedure with agent B is viewed as

$$(3)\quad P_{A\to B}(x_A, x_B, x_1, \ldots x_n) = (\ A(x_A) \mid B(x_B) \mid A_1(x_1) \mid \ldots \mid A_n(x_n)\ ) \setminus C$$

where the A_i are other processes contributing to the protocol. Also, given an enemy process E, the participation of E in the authentication protocol P, as in Eq. 2, is denoted either by $P_{E(A)\to B}$ when E tries to impersonate A in the

eyes of B (for protocols where the instigator wants to authenticate himself), or by $P_{A \to E(B)}$ when E tries to impersonate B in the eyes of A (for protocols where the instigator wants to authenticate B). For the sake of simplicity, this paper considers only one-way authentication protocols where the instigator wants to authenticate himself, thus we shall only consider $P_{E(A) \to B}$ attacks. The other cases, including two-ways authentication protocols, are similar.

For authentication properties, the downgrading actions do not play the same role as in Def. 4.1, the situation being reversed. In Def. 4.5, the set Dwn corresponds rather to a set of admissible attacks from enemy processes. In other words, by viewing any attack attempt on the protocol as interference, we allow enemy processes to cause harmless interference through specific channels. This situation is illustrated in section 6 through the *Woo and Lam one-way authentication protocol* [25].

Definition 4.5 (Preservation of Authenticity) Protocol $P_{A \to B}$ *preserves the authenticity of* A if, for every enemy process E,

$$\forall_{Q \in \mathcal{R}(P_{E(A) \to B})} \quad (Q \setminus Dwn)/\mathcal{O}_B \ \sqsubseteq_{\mathcal{O}_{B,E}} \ Q \setminus Dwn.$$

Once again, this authenticity property may be viewed as follows:

$$\forall_{E\text{: enemy process}} \quad P_{E(A) \to B} \text{ satisfies BNAI}$$

but this time the interpretation of BNAI, given an enemy process E, is such that the high-level actions come from $Hi(E)$ and the low-level actions come from $Hi(B)$ and Lo. As in Def. 4.1, some actions, in fact those from $Hi(A)$ and $Hi(S)$, are not taken into account.

The following unwinding theorem for preservation of authenticity is obtained by applying Theorem 3.3. We omit the proof, which is similar to that of Theorem 4.3.

Theorem 4.6 *Protocol $P_{A \to B}$ preserves the authenticity of A if and only if, for every enemy process E,*

$$\forall_{Q \in \mathcal{R}(P_{E(A) \to B})} \quad Q \setminus Dwn \ \approx_{\mathcal{O}_B} \ Q \setminus (Dwn \ \cup \ Hi(E)).$$

5 The Wide Mouthed Frog Protocol

In [12], the authors proposed a method to detect this attack using a *non deductibility* property and an extension of the *Security Process Algebra* (which is similar to value-passing CCS) called *Crytographic Security Process Algebra*. This extended process algebra introduces encryption and decryption operators in its syntax and deduction rules in its semantics. Our approach, based on Def. 4.1, tends to show that information flow methods can be used without having to extend the process algebra semantics to deal with encryption and decryption, this extension being actually encapsulated in a clever choice of downgrading channels. We back up this assertion with a simplified version of the *Wide Mouthed Frog Protocol* [1] on which a successful attack was revealed in [2].

The Wide Mouthed Frog Protocol is used in order to establish a secure channel between two principals A and B on which A wants to send a confidential message m_A encrypted with a session key k_{AB}. The protocol assumes that A and B share keys k_{AS} and k_{BS} respectively with a trusted third party S (e.g. a server). The protocol consists of the following three messages:

$$\begin{array}{lcl} \text{Message 1:} & A \overset{A,B,\{k_{AB}\}_{k_{AS}}}{\longrightarrow} & S \\ \text{Message 2:} & S \overset{\{A,k_{AB}\}_{k_{BS}}}{\longrightarrow} & B \\ \text{Message 3:} & A \overset{\{m_A\}_{k_{AB}}}{\longrightarrow} & B. \end{array}$$

First, process A sends to S his identifier (A), his counterpart identifier (B) and a fresh key k_{AB} encrypted with a permanent key k_{AS} shared with S that we note $\{k_{AB}\}_{k_{AS}}$. Second, S decrypts $\{k_{AB}\}_{k_{AS}}$ and sends A's identifier and the fresh key k_{AB} to process B encrypted using the shared key k_{BS}. Finally, A sends message m_A to B encrypted with the key k_{AB}. Process B can now decrypt $\{A, k_{AB}\}_{k_{BS}}$ to obtain k_{AB}, and then $\{m_A\}_{k_{AB}}$.

A well known attack on this protocol (reported in [2]) may be pursued by an enemy process E as follows: first, E intercepts *Message 1*, swaps B's identifier with his own and sends it to S. Principal S now believes that A wants to give the session key k_{AB} to E, thus sends $\{A, k_{AB}\}_{k_{ES}}$ to E who can decrypt it to get k_{AB}. Process E may now intercept and decrypt *Message 3* to read the confidential message m_A. This attack will be specified in more details in section 5.2. Before, we need to specify principals A, B and S.

5.1 Protocol Specification

Processes A, B and S are specified using value-passing CCS as follows:

$$\begin{aligned} A(m,k) &= \overline{e_A}(k_{AS},k).cipher_A(\{k\}_{k_{AS}}).\overline{c_1}(A,B,\{k\}_{k_{AS}}). \\ &\quad \overline{e_A}(k,m).cipher_A(\{m\}_k).\overline{c_3}(\{m\}_k) \\ B &= c_2(z).(\ \overline{d_B}(k_{BS},z).read_B((X,u)).c_3(w) + \\ &\quad c_3(w).\overline{d_B}(k_{BS},z).read_B((X,u))\ ).\overline{d_B}(u,w).read_B(v) \\ S &= c_1(X_1,X_2,x).\overline{d_S}(k_{X_1S},x).read_S(y). \\ &\quad \overline{e_S}(k_{X_2S},(X_1,y)).cipher_S(\{(X_1,y)\}_{k_{X_2S}}).\overline{c_2}(\{(X_1,y)\}_{k_{X_2S}}).S \end{aligned}$$

where c_1, c_2 and c_3 are public channels on which messages 1, 2 and 3 are respectively exchanged. We write $C = \{c_1, c_2, c_3\}$. In this particular example, we have $Hi(X) = \{e_X, d_X, read_X\}$. Thus, following the definition of a downgrading process, we have $Hi = \bigcup_X Hi(X)$, $Lo = C = \{c_1, c_2, c_3\}$ and $Dwn = \bigcup_X \{cipher_X\}$. The Wide Mouthed Frog protocol is viewed as follows:

$$P(m_A) = (\ A(m_A, k_{AB}) \mid B \mid S\ ) \setminus C.$$

5.2 Enemy process

An alternative to using the universal quantifier "*for every enemy process E*" from our two properties (Def. 4.1 and Def. 4.5) is to define a "strongest" enemy. This alternative is discussed at the end of this paper, but is irrelevant to this particular example since we set up to prove that the protocol does not preserve the confidentiality of message m_A. To complete such a task, we only have to produce one enemy process for which Def. 4.1 does not hold. For that purpose, we specify in value-passing CCS the enemy process used in [12] which corresponds to the attack mentioned above:

$$E = c_1(X_1, X_2, x).\overline{c_1}(X_1, E, x).c_2(z).(\ \overline{d_E}(k_{ES}, z).read((X, u)).c_3(w) + \\ c_3(w).\overline{d_E}(k_{BS}, z).read((X, u))\ ).\overline{d_E}(u, w).read(v)$$

From Remark 4.2 and this enemy process, we can conclude that the Wide Mouthed Frog protocol $P(m_A)$ does not preserve the confidentiality of message m_A.

6 The Woo and Lam Protocol

To illustrate the authentication property from Def. 4.5, we use the *Woo and Lam one-way authentication protocol* [25]. This particular application illustrates the way that admissible interference permits identification, at the specification level, of possible interferences caused by enemy processes that do not correspond to successful attacks. Such admissible interference, referred to above as admissible attack, has been detected using information flow-based analysis in [8].

This protocol is initiated by a principal A who wants to identify himself with authentication to another principal B where we only assume that both A and B share a permanent encryption/decryption key (noted k_{AS} and k_{BS}) with a trusted third party S (e.g. a server). The protocol is summarized in the following steps:

$$\begin{array}{lcccc}
\text{Message 1:} & A & \xrightarrow{A} & B \\
\text{Message 2:} & B & \xrightarrow{n_B} & A \\
\text{Message 3:} & A & \xrightarrow{\{n_B\}_{k_{AS}}} & B \\
\text{Message 4:} & B & \xrightarrow{\{A,\{n_B\}_{k_{AS}}\}_{k_{BS}}} & S \\
\text{Message 5:} & S & \xrightarrow{\{n_B\}_{k_{BS}}} & B.
\end{array}$$

First, A initiates the protocol by sending his identifier to B, and B responds by sending a fresh nonce n_B to A. The latter then sends back n_B encrypted with key k_{AS}. Principal B can now proceed to authenticate A with the help of S by sending A's identifier and the last message received from A, both encrypted with key k_{BS}. The trusted third party S decrypts this message

from B using k_{BS}, then decrypts $\{n_B\}_{k_{AS}}$ using k_{AS} and, finally sends n_B back to B encrypted with k_{BS}. Once this last message has been decrypted, B only has to verify whether or not the resulting value corresponds to its initial nonce n_B to approve A's authentication.

In section 6.2, we are interested in the attack on this protocol that was reported in [26]. In this particular attack, principal A initiates the protocol with enemy process E which forwards all information received from A to another principal B in order for E to impersonate A.

6.1 Protocol Specification

In order to specify Woo and Lam's protocol using value-passing CCS, we define principals A (instigator), B (respondent) and S (server) as follows:

$$\begin{aligned}
A(X_B) &= \overline{init_A}(X_B).\overline{c_{1X_B}}(A).c_{2A}(x).\overline{e_A}(k_{AS},x).cipher_A(\{x\}_{k_{AS}}).\\
&\quad \overline{commit_A}(X_B,x).\overline{c_{3X_B}}(\{x\}_{k_{AS}})\\
B(n) &= c_{1B}(X_A).\overline{request_B}(X_A).\overline{c_{2X_A}}(n).c_{3B}(y).\overline{e_B}(k_{BS},(X_A,y)).\\
&\quad cipher_B(\{(X_A,y)\}_{k_{BS}}).\overline{c_4}(\{(X_A,y)\}_{k_{BS}}).c_{5B}(w).\overline{d_B}(k_{BS},w).\\
&\quad read_B(u).[u=n]\ \overline{auth_B}(X_A)\\
S(X_B) &= c_4(z_1).\overline{d_S}(k_{X_BS},z_1).read_S((X_A,z_2)).\overline{d_S}(k_{X_AS},z_2).read_S(z_3).\\
&\quad \overline{e_S}(k_{X_BS},z_3).cipher_S(\{z_3\}_{k_{X_BS}}).\overline{c_{5X_B}}(\{z_3\}_{k_{X_BS}})
\end{aligned}$$

where we use notation established above for encryption channels and decryption channels. We use c_{iX} to denote the public channel used to send the ith message intended for X. We also added the following private channels:

- $\overline{init_X}(X')$: to indicate that X wants to initiate the protocol with X';
- $\overline{request_X}(X')$: to indicate that X (believes he) just received a request to execute the protocol form X';
- $\overline{commit_X}(X',x)$: to indicate that X is committed to identify himself to X' with authentication using nonce x;
- $\overline{auth_X}(X')$: to indicate that X (believes he) has authenticated X'.

The Woo and Lam protocol can be viewed as follows:

$$P_{A\to B} = (\ A(B) \mid B(n_B) \mid S(B)\ ) \setminus C$$

where $C = \bigcup_X \{c_{1X}, c_{2X}, c_{3X}, c_4, c_{5X}\}$ and we put $Hi(X) = \bigcup_{X'} \{e_X, cipher_X, d_X, read_X, init_X(X'), request_X(X'), commit_X(X',x), auth_X(X')\}$ and $Lo = C$. Note that we have not considered downgrading actions yet, since such actions are interpreted as admissible interference caused by an enemy process and hence they only appear in such processes, as we shall see next.

6.2 Enemy Process

As in the previous section, we have postponed the task of constructing a "greatest enemy process" in future works and concentrate on the flaw revealed by Abadi in the Woo and Lam protocol [26] using BNAI. More precisely, we see that the Woo and Lam one-way authentication protocol specified as $P_{A\to B}$ does not preserve the authenticity of A.

To achieve this, we consider the following enemy process executing the attack reported in [26] and [8]:

$$E(X_B) = c_{1E}(X_A).\overline{request_E}(X_A).dwn_E.\overline{init_E}(X_B).dwn_E.\overline{c_{1X_B}}(X_A).$$
$$c_{2X_A}(x).\overline{c_{2X_A}}(x).c_{3E}(y).\overline{commit_E}(X_B).\overline{c_{3X_B}}(y)$$

where we consider any action of type $\overline{request_E}(X)$ or $\overline{init_E}(X)$ as admissible interference from E, putting $Dwn = \{dwn_E\}$.

Thus, the attack on the Woo and Lam protocol is expressed as follows:

$$P_{E(A)\to B} = (\ A(E) \mid B(n_B) \mid S(B) \mid E(B)\ ) \setminus C$$

where principal A tries to authenticate himself toward E, but the latter uses data received from A to steal his identity and successfully authenticate himself as A toward B. In the end, B believes that E is A.

Using Theorem 4.6, we can see that $P_{A\to B}$ does not preserve the authenticity of A since

$$Q \setminus Dwn \ \not\approx_{\mathcal{O}_B}\ Q \setminus (Dwn \ \cup \ Hi(E))$$

for some $Q \in \mathcal{R}(P_{E(A)\to B})$. Such Q is given by any derivative that can execute a computation of $\overline{request_B}(A).\overline{commit_A}(E).\overline{commit_E}(B).\overline{auth_B}(A)$. This sequence of actions becomes $\overline{request_B}(A).\tau.\tau.\overline{auth_B}(A)$ in $(Q \setminus Dwn)/\mathcal{O}_B$, but the same sequence becomes $\overline{request_B}(A).\tau$ in $(Q\setminus(Dwn \cup Hi(E))/\mathcal{O}_B$. Thus, we may say that $(Q \setminus Dwn)/\mathcal{O}_B \ \not\approx (Q \setminus (Dwn \ \cup \ Hi(E)))/\mathcal{O}_B$ which leads us to our conclusion.

Note that a similar approach using non interference was proposed in [8]. The authors have to filter interference, after their analysis, that does not correspond to attacks such as the trace

$$\overline{init_A}(E).\overline{request_E}(A).\overline{init_E}(B).\overline{request_B}(A).\overline{commit_A}(E).$$

Admissible interference allows specification of these harmless attacks and only failures caused by successful attacks, to be obtained from any analysis of the protocol. Also, by identifying such admissible interference before initiating an automatic analysis of a security protocol, results are gained with precision and clarity. A cost savings on the software design process might also be expected.

7 Final Remarks and Related Works

The main contributions of this paper are a bisimulation-based generalization of trace-based admissible interference initially proposed in [19], its correspond-

ing unwinding theorem (Theorem 3.3) and a compositionality theorem (Theorem 3.4) w.r.t. the main constructors of concurrent processes. Moreover, as a non-trivial application of BNAI, it is proposed in Section 4 a new approach to analyze cryptoprotocols. This approach extends the approach based on non interference presented in [8,11,12,9]. Confidentiality and authentication are defined in terms of BNAI and their respective bisimulation-based proof method (Theorems 4.3 and 4.6 respectively) are derived. Its main advantage over a non interference-based approach is to reveal flaws with more efficiency by discarding harmless attacks earlier in the protocol's design process and to permit the use of a general purpose process algebra, instead of specialized process algebra extended with with encryption-decryption primitives to cope with admissible interference caused by the cryptosystem. This method has been illustrated in detail in two case studies: the Wide Mouthed Frog protocol (Section 5) and the Woo and Lam one-way authentication protocol (Section 6).

In addition to the papers mentioned above, the process algebraic approach to cryptographic protocols has also been followed in [21,16,22] that consider model-checking of security protocols in a CSP-based framework. This approach requires explicitly designing a specific (powerful enough) intruder. Of course, there is always a certain amount of arbitrarity in determining this intruder and any modification of the intruder would require a new analysis. In our paper, a more radical approach is taken: the intruder may be *any* process that can be defined in CCS. We postpone the discussion about this crucial issue to the end of this section, and mention some promising research threads.

We are investigating more general properties of intransitive non interference for processes, inspired by Pinsky's study [20]. It appears indeed that the algorithm presented by Pinsky to construct a minimal equivalence and its associated unwinding condition for a downgrading policy, can be thought of as an algorithm to construct the appropriate bisimulation.

Motivated by the ability of the π-calculus, its variants and extensions to model mobility more accurately and hence, secure distributed applications over the Internet, we believe that admissible interference and more generally intransitive non interference should be characterized in terms of such calculi. A further step will be then to extend our compositional and complete (at least for finite-state processes) information flow method to the analysis of cryptographic protocols for such calculi.

In the last few years, many approaches based on the π-calculus, have been proposed to analyze security protocols. Below we would like to highlight three among those we intend to focus our attention on, in view of further developments: the Abadi-Gordon's Spi-calculus [3] and its sound but incomplete *framed bisimulation*-based proof method [2], the Boreale *et al.*'s variant of the Spi-calculus [5] with its sound and complete *barbed bisimulation*-based proof method and the control flow analysis for the π-calculus presented in [4].

Although they are not based on information flow approach to secrecy, the

Spi-calculus approaches are inspiring for further developments of our method, particularly in the way that the problem of the "most powerful intruder" briefly mentioned above, is overcome. In this paragraph, we discuss this major issue. A security property should be satisfied even in presence of a hostile environment. Also, it should be resistant to every potential attacker and, checking this condition is generally intractable. The Spi-calculus overcomes this problem by representing security properties as weakened form of testing equivalence. Let $P(M)$ be a process P processing a piece of data m. From the Spi-calculus point of view, P preserves secrecy of m if there is no test with the capability to discriminate $P(m)$ from $P(m')$, for every m'. A test nicely formalizes the idea of a generic experiment or observation that another Spi-process (a potential attacker) might perform on P. So P and Q are testing equivalent if there exists no attacker powerful enough to discriminate them. Also, Abadi-Gordon's definition [3] suffers from quantification over all possible contexts. In [5], it is designed as an enriched labeled transition system, used to define a weak bisimulation equivalence, that avoids quantification over contexts and leads to a complete proof method. Further research is required for a fuller understanding of these notions and for tailoring up information flow techniques to reason over them. But we apprehend already that introducing encryption-decryption primitives in the π-calculus leads to a bisimulation method that has to deal with additional semantic rules. Moreover, we conjecture that these rules can be captured by a right interpretation of downgrading and an adequate observation criterion of this enriched labeled transition system in order to admit any interference caused by the inevitable correlation between a ciphertext and its related text. More recently, an information flow approach based on the π-calculus has been proposed with application to the control flow analysis of cryptoprotocols in [4]. We conjecture that the simple security properties established by the authors, namely the *no leaks* and the *no read-up/no write-down* properties, do not allow to analyze subliminal channels in authentication protocols, contrarily to information flow properties like non interference and admissible interference.

Finally, from a completely different point of view, we are trying to exploit the well-known result establishing decidability of bisimulation over some classes of infinite-state processes, e.g. totally normed Basic Process Algebra (BPA) [14], and hence, over pushdown automata [24]. It is indeed an attractive avenue to address the "most powerful intruder" as the process using a queue or a stack as an infinite memory and having access to any public channels, its own private channels allowing him to encrypt and decrypt and any other initial data such as shared encryption keys, nonces and so on.

References

[1] M. Abadi, M. Burrows, and R. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, 1990.

[2] M. Abadi and A. D. Gordon. Reasoning about cryptographic protocols in the spi calculus. In *CONCUR'97: Concurrency Theory*, volume 1243, pages 59–73, Berlin, Germany, 1997. Springer-Verlag.

[3] M. Abadi and A. D. Gordon. A bisimulation method for cryptographic protocols. *Nordic Journal of Computing*, 5(4):267–303, Winter 1998.

[4] C. Bodei, P. Degano, F. Nielson, and H. Nielson. Static analysis for the π- calculus with applications to security. *Information and Computation*, (to appear) 2001.

[5] M. Boreale, R. De Nicola, and R. Pugliese. Proof techniques for cryptographic processes. In *Logic in Computer Science*, pages 157–166, 1999.

[6] G. Boudol. Notes on algebraic calculi of processes. In *Logic and Models of Concurrent Systems*, NATO ASI Series F-13, pages 261–303. Springer, 1985.

[7] R. Cleaveland, J. Parrow, and B. Steffen. The concurrency workbench: a semantics based tool for the verification of concurrent processes. *ACM Transactions on Programming Languages and Systems*, 15(1):36–72, January 1993.

[8] A. Durante, R. Focardi, and R. Gorrieri. CVS: A compiler for the analysis of cryptographic protocols. In *Proceedings of 12th IEEE Computer Security Foundations Workshop*. IEEE Computer Society, June 1999.

[9] R. Focardi, A. Ghelli, and R. Gorrieri. Using non interference for the analysis of security protocols. In H. Orman and C. Meadows, editors, *Proceeding of the DIMACS Workshop on Design and Formal Verification of Security Protocols*, Rutgers University, September 1997. DIMACS Center.

[10] R. Focardi and R. Gorrieri. A classification of security properties for process algebras. *J. of Computer Security*, 3(1):5–33, 1994/1995.

[11] R. Focardi, R. Gorrieri, and F. Martinelli. Secrecy in security protocols as non interference. In S. Schneider and P. Ryan, editors, *Proceedings of DERA/RHUL Workshop on Secure Architectures and Information Flow*, volume 32. Elsevier ENTCS, 2000.

[12] R. Focardi and F. Martinelli. A uniform approach for the definition of security properties. In *Proc. of World Congress on Formal Methods (FM'99)*, Springer, LNCS 1708, pages 794–813, 1999.

[13] J.A. Goguen and J. Meseguer. Unwinding and inference control. In *Proceedings of the IEEE Symp. on Research in Security and Privacy*, pages 75–285, Oakland, CA, 1984. IEEE Computer Society.

[14] Y. Hirshfeld and F. Moller. Decidability results in automata and process theory. In F. Moller and G. Birthwistle, editors, *Logic for Concurrency: Structure versus Automata*, volume 1043 of *LNCS*, pages 102–148. Springer, 1996.

[15] P. Kannellakis and S. Smolka. Ccs expressions, finite state processes and three problems of equivalence. *Information and Computation*, 86:333–354, 1990.

[16] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Proc. of TACAS'96*, volume 1055 of *LNCS*, pages 147–166. Springer-Verlag, 1996.

[17] J. McLean. Security models. In J. Marciniak, editor, *Encyclopedia of Software Engineering*, pages 1136–1145. John Wyley & Sons, 1994.

[18] R. Milner. *Communication and concurrency*. Prentice-Hall, 1989.

[19] J. Mullins. Nondeterministic admissible interference. *Journal of Universal Computer Science*, 6(11):1054–1070, 2000.

[20] S. Pinsky. Absorbing covers and intransitive non-interference. In *Proceedings of the IEEE Symp. on Research in Security and Privacy*, pages 102–113, Oakland, CA, May 1995. IEEE Computer Society.

[21] A.W. Roscoe. Modelling and verifying key-exchange using CSP and FDR. In *8th Computer Security Foundation Workshop*. IEEE Computer Society Press, 1995.

[22] S. Schneider. Verifying authentication protocols in CSP. *IEEE Transactions of Software Engineering*, 24(8):447–479, 1998.

[23] C. Stirling. Decidability of bisimulation equivalence for pushdown processes. unpublished, 2000.

[24] C. Stirling. Decidability of DPDA equivalence. Technical Report LFCS99-411, University of Edinburgh, 2000. to appear in Theoretical Computer Science.

[25] T. Woo and S. Lam. Authentication for distributed systems. *IEEE Computer*, 25(1):39–52, January 1992.

[26] T. Y. C. Woo and S. S. Lam. A lesson in authentication protocol design. *ACM Operating Systems Review*, 28(3), 1994.

A Theorem's proof

A.1 Proof of the Unwinding Theorem for BNAI

Proof of Theorem 3.3 Given $Q = P' \setminus Dwn$ for $P' \in \mathcal{R}(P)$, it is sufficient to prove that

$$Q \sqsubseteq_{\mathcal{O}_{Lo}} Q \setminus Hi \tag{A.1}$$

since any $\mathcal{O}_{Lo}$-simulation of Q by $Q \setminus Hi$ is actually a $\mathcal{O}_{Lo}$-bisimulation.

By definition of $\mathcal{O}_{Hi}$-simulation and since $(Q/\mathcal{O}_{Lo})/\mathcal{O}_{Hi} \approx Q/\mathcal{O}_{Lo}$, we have

$$Q/\mathcal{O}_{Lo} \sqsubseteq_{\mathcal{O}_{Hi}} Q \Longleftrightarrow Q/\mathcal{O}_{Lo} \sqsubseteq Q/\mathcal{O}_{Hi} \tag{A.2}$$

by Prop. 2.4, and it is not difficult to see that

$$Q/\mathcal{O}_{Lo} \sqsubseteq Q/\mathcal{O}_{Hi} \Longleftrightarrow Q/\mathcal{O}_{Lo} \sqsubseteq (Q/\mathcal{O}_{Hi}) \setminus Hi. \tag{A.3}$$

Indeed, given a $(Q/\mathcal{O}_{Lo})$-transition $Q_1 \xrightarrow{\alpha} Q_2$ (hence with $Q_1 \in \mathcal{R}(Q)$ and $\alpha \in Act \setminus Hi$), $Q_1 \xrightarrow{\alpha} Q_2$ is a $Q/\mathcal{O}_{Hi}$-transition if and only if it is a $((Q/\mathcal{O}_{Hi}) \setminus Hi)$-transition. Moreover, we have

$$(Q/\mathcal{O}_{Hi}) \setminus Hi \approx (Q \setminus Hi)/\mathcal{O}_{Hi} \tag{A.4}$$

$$\approx (Q \setminus Hi)/\mathcal{O}_{Lo} \tag{A.5}$$

Hence, putting Eqs. A.2- A.5 together, we obtain:

$$Q/\mathcal{O}_{Lo} \sqsubseteq (Q \setminus Hi)/\mathcal{O}_{Lo} \tag{A.6}$$

and, by Prop. 2.4, Eq. A.6 is equivalent to Eq. A.1. □

A.2 Proof of the Compositionality Theorem for BNAI

The next proposition, proved in [18], shows that strong bisimulation is a congruence with respect to the concurrent and restriction operators, and that there is a weak form of distributivity of the restriction operator over the concurrent one.

Proposition A.1 *If $P_1 \approx Q_1$ and $P_2 \approx Q_2$, then*

(i) $P_1|P_2 \approx Q_1|Q_2$

(ii) $P_1 \setminus L \approx Q_1 \setminus L$

(iii) *If P_1 and P_2 may not synchronize on actions in L, then*

$$(P_1|P_2) \setminus L \approx (Q_1 \setminus L)|(Q_2 \setminus L).$$

The proof of Theorem 3.4 requires the following lemma stating that the functional composition of the restriction to Dwn and of a quotient with $\mathcal{O}_{Hi}$ is distributive over the concurrent composition.

Lemma A.2 *If processes P and Q may not synchronize on downgrading actions, then*

$$((P|Q) \setminus Dwn)/\mathcal{O}_{Lo} \approx ((P \setminus Dwn)/\mathcal{O}_{Lo})|((Q \setminus Dwn)/\mathcal{O}_{Lo})$$

Proof. It is sufficient to show that

$$((P|Q) \setminus Dwn)/\mathcal{O}_{Lo} \sqsubseteq ((P \setminus Dwn)/\mathcal{O}_{Lo})|((Q \setminus Dwn)/\mathcal{O}_{Lo})$$

because any simulation of $((P|Q) \setminus Dwn)/\mathcal{O}_{Lo}$ by $((P \setminus Dwn)/\mathcal{O}_{Lo})|((Q \setminus Dwn)/\mathcal{O}_{Lo})$ is actually a bisimulation. This results trivially from Prop. A.1.

For the $\sqsubseteq$ simulation, we proceed by structural induction on the concurrent composition rules. The only difficult case is the one raised from a τ transition by high-level action synchronization resulting from application of the `Synchronization` rule. Let $P_1 \in \mathcal{R}(P)$ and $Q_1 \in \mathcal{R}(Q)$ be such that $P_1|Q_1 \xrightarrow{\tau} P_2|Q_2$, a $P|Q$-transition issued from the P-transition $P_1 \xrightarrow{\alpha} P_2$ and the Q-transition $Q_1 \xrightarrow{\overline{\alpha}} Q_2$ with $\alpha \in Hi$. This results in the $((P \setminus Dwn)/\mathcal{O}_{Lo}))$-transition $P_1 \xrightarrow{\tau} P_2$ and the $((Q \setminus Dwn)/\mathcal{O}_{Lo}))$-transition $Q_1 \xrightarrow{\tau} Q_2$ to obtain the $(((P \setminus Dwn)/\mathcal{O}_{Lo}))|((Q \setminus Dwn)/\mathcal{O}_{Lo})))$-transition $P_1|Q_1 \xrightarrow{\tau} P_2|Q_2$. □

Proof of Theorem 3.4

(i) Given $P' \in \mathcal{R}(P)$ and $Q = P' \setminus Dwn$, then in view of Theorem 3.3, it suffices to prove:

$$Q \approx_{\mathcal{O}_{Lo}} Q \setminus Hi \implies Q \setminus L \approx_{\mathcal{O}_{Lo}} Q \setminus (Hi \cup L)$$

We have:

$$\begin{aligned} Q \approx_{\mathcal{O}_{Lo}} Q \setminus Hi &\iff Q/\mathcal{O}_{Lo} \approx (Q \setminus Hi)/\mathcal{O}_{Lo} \\ &\quad \text{by Prop. 2.4} \\ &\implies (Q/\mathcal{O}_{Lo}) \setminus L \approx ((Q \setminus Hi)/\mathcal{O}_{Lo}) \setminus L \\ &\quad \text{by Prop. A.1} \\ &\implies (Q \setminus L)/\mathcal{O}_{Lo} \approx (Q \setminus (Hi \cup L))/\mathcal{O}_{Lo} \\ &\iff Q \setminus L \approx_{\mathcal{O}_{Lo}} Q \setminus (Hi \cup L) \\ &\quad \text{by Prop. 2.4.} \end{aligned}$$

(ii) Let $P'|Q' \in \mathcal{R}(P|Q)$. It is sufficient to show that

(A.7) $$(P'|Q') \setminus Dwn \sqsubseteq_{\mathcal{O}_{Lo}} ((P'|Q') \setminus (Hi \cup Dwn))$$

in view of Theorem 3.3 and the fact that any $\mathcal{O}_{Lo}$-simulation of $(P'|Q') \setminus Dwn$ by $((P'|Q') \setminus (Hi \cup Dwn))$ is actually a $\mathcal{O}_{Lo}$-bisimulation. By Prop. 2.4, Eq A.7 is equivalent to:

$$((P'|Q') \setminus Dwn)/\mathcal{O}_{Lo} \sqsubseteq ((P'|Q') \setminus (Hi \cup Dwn))/\mathcal{O}_{Lo}.$$

We have:

$$\begin{aligned} ((P'|Q') \setminus Dwn)/\mathcal{O}_{Lo} &\approx ((P' \setminus Dwn)/\mathcal{O}_{Lo})|((Q' \setminus Dwn)/\mathcal{O}_{Lo}) \\ &\quad \text{by Lemma A.2} \\ &\approx ((P' \setminus (Dwn \cup Hi))/\mathcal{O}_{Lo})|((Q' \setminus (Dwn \cup Hi))/\mathcal{O}_{Lo}) \\ &\quad \text{by Prop. A.1 and Theorem 3.3} \\ &\approx ((P' \setminus (Dwn \cup Hi))|(Q' \setminus (Dwn \cup Hi)))/\mathcal{O}_{Lo} \\ &\quad \text{by Lemma A.2} \\ &\sqsubseteq ((P'|Q') \setminus (Dwn \cup Hi))/\mathcal{O}_{Lo}. \end{aligned}$$

□

A.3 Preservation of Confidentiality

Proof of Theorem 4.3 Since both statements use the same domain for enemy processes, then, given an enemy process E and a derivative $P'_E \in \mathcal{R}(P_E)$, we only have to show that

$$(P'_E(m) \setminus Dwn)/\mathcal{O}_E \sqsubseteq_{\mathcal{O}_m} P'_E(m) \setminus Dwn$$

if and only if

$$P'_E(m) \setminus Dwn \approx_{\mathcal{O}_E} P'_E(m) \setminus (Dwn \cup Act(m)).$$

Let E be an enemy process, $P'_E \in \mathcal{R}(P_E)$ a derivative and $Q = P'_E(m) \setminus Dwn$.

As in the proof of Theorem 3.3, we see that

$$\begin{aligned} Q/\mathcal{O}_E \ \sqsubseteq_{\mathcal{O}_m} \ Q &\iff Q/\mathcal{O}_E \ \sqsubseteq \ Q/\mathcal{O}_m \\ &\iff Q/\mathcal{O}_E \ \sqsubseteq \ (Q/\mathcal{O}_m) \setminus Act(m) \\ &\iff Q/\mathcal{O}_E \ \sqsubseteq \ (Q \setminus Act(m))/\mathcal{O}_E \\ &\iff Q \ \sqsubseteq_{\mathcal{O}_E} \ Q \setminus Act(m) \\ &\iff Q \ \approx_{\mathcal{O}_E} \ Q \setminus Act(m). \end{aligned}$$

□